UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/028,858	12/19/2001	Shivnandan D. Kaushik	42390P13163	4672

7590 10/04/2005

John P. Ward
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
VO, TIM T	
ART UNIT	**PAPER NUMBER**
2112	

DATE MAILED: 10/04/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/028,858
Filing Date: December 19, 2001
Appellant(s): KAUSHIK ET AL.

Mark C. Van Ness
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 07/11/2005 appealing from the Office action mailed 02/07/2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal only claims 1-10 and 16-34 are appealed and claims 10-14 is not appealed.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,282,596	Bealkowski	8-2001
6,587,909	Olarig et al.	7-2003

Part III DETAILED ACTION

Notice to Applicant(s)

This application has been examined. Claims 1-10, 12-14 and 16-34 are pending.

Claims 1-10, 12-14 and 16-34 maintain rejected under 35 U.S.C. § **102**(**e**) as being anticipated by Bealkowski et al. patent number 6,282,596.

Examiner introduces a new reference Olarig et al. 6,587,909 to reject broadest claim 8 under 35 U.S.C. § **102**(**e**) as being anticipated by Olarig et al. patent number 6,587,909 for a new ground of rejection with the director of TC 2100 approval.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:
A person shall be entitled to a patent unless --

> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

Claims 1, 7-9 are is rejected under 35 U.S.C. § **102**(**e**) as being anticipated by Olarig et al. patent number 6,587,909 referred hereinafter "Olarig".

As for claim 8, Olarig teaches a method of adding memory to a running computing device (see figure 1, computer system 10, memory modules 14 and column

6 lines 50-57, wherein additional memory modules are added by hot-plugging to the computer system 10), comprising:

Identifying memory of a hot plug module in response to the hot plug module being physically coupled to the running computing device (see column 5 lines 1-23, wherein the System Control Interrupt (SCI) generates interrupt to the ACPI driver for "hot add" memory event. The ACPI driver responds to the "hot add" memory event by performing the 5 tasks, wherein the first task is to determine if memory has been added or removed and if the has been added, determine whether if it is a "hot addition" or a "hot replacement". This task is equivalent to what is claimed by identifying a memory module, which is being added or removed from the computer system 10. Furthermore, tasks 2-5 performing the memory configuration for the "hot add" memory event).

Adding the identifying memory to the hot plug module to a memory pool of the running computing device (see column 3 lines 15-20, wherein the memory modules are added to the computer system as claimed).

As for claims 1, 7 and 9, Olarig teaches a method of adding one or more caching agents to a running computing device (see figure 1, memory modules 14, computer system 10), comprising:

Identifying the one or more caching agents provided by a hot plug module in response to the hot plug module being physically coupled to the running computing device (see column 5 lines 1-23, wherein the System Control Interrupt (SCI) generates interrupt to the ACPI driver for "hot add" memory event. The ACPI driver responds to the "hot add" memory event by performing the 5 tasks, wherein the first task is to

determine if memory has been added or removed and if the has been added, determine whether if it is a "hot addition" or a "hot replacement". This task is equivalent to what is claimed by identifying a memory module, which is being added or removed from the computer system 10. Furthermore, tasks 2-5 performing the memory configuration for the "hot add" memory event).

Adding the identified caching agents of the hot plug module to a resource pool of the running computing device (see column 3 lines 15-20, wherein the memory modules are added to the computer system as claimed).

Enabling communication interface of the hot plug module to establish a communication link with the running computing device (see column 5 lines 1-23, wherein the 5 tasks performing the configuration for establishing communication between the newly added memory module and the computer system 10).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:
A person shall be entitled to a patent unless --

> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

Claims 1-10, 12-14 and 16-34 are rejected under 35 U.S.C. § **102(e)** as being anticipated by Bealkowski et al. patent number 6,282,596 referred hereinafter "Bealkowski".

As for claims 1 and 8, Bealkowski teaches a method of adding one or more caching agents to a running computing device (see figure 1-3, hot plug processor cards 11, 20, 30 and column 3 lines 9-11, column 4 lines 4-11, wherein each processor card is hot plug and each processor card comprises a cache), comprising identifying the one or more caching agents provided by a hot plug module in response to the hot plug module being physically coupled to the running computing device (see figure 4 and column 7 line 51 to column 8 line 31, wherein each processor is identified with an agent ID and the hot plug controller uses the agent ID identify which processor card is removed or inserted to the computer system) ; and adding the identified caching agents of the hot plug module to a resource pool of the running computing device (see figure 2, service processor 31 and column 5 lines 31-60, wherein the service processor 31 controls the hot plug controller and also monitoring the events in the computer such as insertion or removal and then stores such information to its own associated memory and controller routine. Further, column 3 lines 35-41, teaches integrated processor cards which including cache into the data processing. Therefore, the caches of the processor cards are considered adding memory to the data processing as well); and enabling a communication interface of the hot plug module to establish a communication link with the running computing device (see figures 1-3, processor cards 11, 20, 30 are hot plugged to the data processing system 10 via the CPU connectors. Further, data

processing system 10 supports hot plugging and initializing processor cards for communication between the newly added processor card and the processing system 10 via the CPU connectors as discloses in column 3 lines 1-15 and lines 35-41). As for claims 2 and 9, Bealkowski further teaches comprising enabling a communication interface of the running computing device that is associated with the hot plug module in response to determining that the hot plug module has been physically coupled to the running computing device (see figure 4, wherein the flow chart describes the processor for establishing communication between the processor cards and the computer system).

As for claims 3 and 10, Bealkowski teaches performing a self test of the hot plug module, and in response to passing the self test, enabling the communication interface of the hot plug module to establish a communication link with the communication interface of the running computing device (see column 8 lines 31-39).

As for claim 4, Bealkowski teaches initializing the hot plug module, and after initializing the hot plug module, enabling the communication interface of the hot plug module to establish a communication link with the communication interface of the running computing device (see column 9 lines 10 lines 4-10).

As for claim 5, Bealkowski teaches wherein adding comprises adding one or more memory caching processors of the identified caching agents to a processor pool of the running system (see column 5 lines 53-60).

As for claim 6, Bealkowski teaches wherein adding comprises adding one or more memory caching input/output hubs of the identified caching agents to an input/output pool of the running system (see figure 1, CPU connector 14).

As for claim 7, Bealkowski teaches identifying memory of the hot plug module in response to the hot plug module being physically coupled to the running computing device; and adding the identified memory of the hot plug module to a memory pool of the running computing device (see column 8 lines 13-25).

As for claim 12, Bealkowski teaches in response to the hot plug removal request, providing an indication that a hot plug removal is in progress (see figure 5 and column 9 lines 33-51).

As for claim 13, Bealkowski teaches after removing the identified resources from the running computing device, providing an indication that the hot plug module may be removed; and disabling the communication interface of the running computing device to isolate the hot plug module from the running computing device (see figure 5 and column 9 lines 33-51).

As for claim 14, Bealkowski teaches a method of removing a hot plug module comprising one or more memory caches from a running computing device (see figure 1-3, hot plug processor cards 11, 20, 30 and column 3 lines 9-11, column 4 lines 4-11, wherein each processor card is hot plug and each processor card comprises a cache), comprising identifying resources of the hot plug module in response to a hot plug removal request (see figure 5 and column 9 lines 33-51); causing the hot plug module to write back modified cache lines of the one or more memory caches to the running computing device (see column 5 lines 53-60); and removing the identified resources from respective resource pools of the running computing device (see column 9 lines 33-51).

Bealkowski teaches waiting a predetermined time for pending transactions associated with hot plug module to complete; and disabling the communication interface of the running computing device to isolate the hot plug module from the running computing device after waiting the predetermined time (see figure 4, step 130 and column 9 lines 9-32).

As for claim 16, Bealkowski teaches readable medium for interrupt processing, comprising a plurality of instructions that in response to being executed result in a computing device in response to a hot plug interrupt examining a plurality of interface control registers associated with a plurality of communication interfaces for communicating with a plurality of hot plug modules having caching agents (see figure 4 and column 7 lines 51-65); and identifying which of a plurality of hot plug events caused the hot plug interrupt based upon the plurality of interface control registers (see column 8 lines 13-29).

As for claim 17, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device determining whether hot plug addition or hot plug removal has been requested for a hot plug module of the plurality of hot plug modules based upon an interface control register of the plurality of interface control registers that is associated with the hot plug module (see figures 4-5, wherein figure 4 discloses hot plug addition and figure 5 discloses hot plug removal).

As for claim 18, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device determining that hot plug removal has been requested for the hot plug module in response to the interface

control register associated with the hot plug module indicating a pending hot plug interrupt and a joined state for the hot plug module (see figures 4-5, step 130 and column 9 lines 9-32).

As for claim 19, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device causing the hot plug module to write back modified cache lines to the running computing device; and removing resources of the hot plug module from the computing device (see column 5 lines 53-60).

As for claim 20, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device determining that hot plug removal has been requested for the hot plug module (see figure 5); waiting a predetermined time for pending transactions associated with hot plug module to complete (see figure 4, step 130 and column 9 lines 9-32); and disabling the communication interface of the computing device to isolate the hot plug module from the computing device after waiting the predetermined time (see figure 5 step 146).

As for claim 21, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device determining that hot plug removal has been requested for the hot plug module (see figure 5); and disabling the communication interface of the computing device to isolate the hot plug module from the computing device after determining that all transactions associated with the hot plug module have completed (see figure 5 step 146).

As for claim 22, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the computing device determining that hot plug addition has been requested for the hot plug module in response to the interface control register indicating that the associated communication interface is disabled, that a module is coupled to the associated communication interface, and that the associated communication interface is in a no module present state (see figure 4 and column 8 lines 1-6).

As for claim 23, Bealkowski teaches, wherein the plurality of instructions in response to being executed further result in the processor adding resources of the hot plug module to the running computing device in response to determining that hot plug addition has been requested for the hot plug module and that no other hot plug addition is in progress (see figure 4 and column 7 lines 51-65).

As for claim 24, Bealkowski teaches a hot plug module comprising a coupler for detachably coupling the hot plug module to a running computing device (see figure 1-3, hot plug processor cards 11, 20, 30 and column 3 lines 9-11, column 4 lines 4-11, wherein each processor card is hot plug to the CPU connector 14); a communication interface to establish a communication link with the running computing device via the coupler in response to being enabled and to de-establish the communication link in response to being disabled (see figures 4-5, wherein figure 4 discloses enabling communication for addition of processor card and figure 5 discloses de-establish communication for removal of processor card); an interface control register associated with the communication interface to indicate and control whether the communication

interface is enabled or disabled (see figure 2, hot plug controller 70), and a processor and associated memory cache, the processor to program the interface control register to enable and disable the communication interface based upon whether the hot plug module is ready to join the running computing device (see processor card 30, 20, 11 and figures 4-5 disclosing enable communication with added processor and figure 5 disclose disable communication with removed processor card).

As for claim 25, Bealkowski teaches wherein the processor writes back modified cache lines of the memory cache to the running computing device in response to hot plug removal being requested for the hot plug module (see column 5 lines 53-60).

As for claim 26, Bealkowski teaches, wherein the hot plug module comprises a status indicator that indicates a hot plug status for the hot plug module (see LED service indicator 72 and column 5 lines 41-52).

As for claim 27, Bealkowski teaches, wherein the processor performs a self test of the hot plug module in response to hot plug addition being requested for the hot plug module, and enables the communication interface in response to determining that the hot plug module passed the self test (see column 8 lines 31-56).

As for claim 28, Bealkowski teaches a computing device comprising, a midplane comprising a coupler and a hot plug interface to track a state associated with the coupler (see figure 1); a hot plug module comprising a coupler to detachably couple the hot plug module to the coupler of the midplane and resources coupled to the coupler of the hot plug module via a hot plug interface of the hot plug module (see figure 1, processor cards are connected to the CPU connectors 14 to the system bus 18), the hot

plug module to update the state of the hot plug interface of the midplane to indicate when the resources are ready to join the computing device (see column 5 lines 41-52); and a processor coupled to the hot plug interface of the midplane (see figure 1, service processor 31), the processor to add the resources to the computing device without rebooting in response to determining that the hot plug interface of the midplane indicates the resources are ready to join (see column 1 lines 57-58).

As for claim 29, Bealkowski teaches, wherein the midplane comprises a hot plug monitor that provides the hot plug interface of the midplane with a signal indicative of whether the coupler of the hot plug module has been coupled to the coupler of the midplane, and the processor programs the hot plug interface of the midplane to generate a hot plug interrupt in response to a change in the signal that is indicative of whether the coupler of the hot plug module has been coupled to the coupler of the midplane (see column 8 lines 1-5).

As for claim 30, Bealkowski teaches, wherein the hot plug interface of the midplane detects whether framing packets are received from the hot plug interface of the hot plug module, and the processor programs the hot plug interface of the midplane to generate a hot plug interrupt in response to a change in receipt of framing packets (see column 7 line 52 to column 8 line 29).

As for claim 31, Bealkowski teaches a midplane of a computing device, comprising a plurality of couplers to detachably couple hot plug modules to the midplane (see figure 1, CPU connector 14); at least one switch to interconnect the plurality of couplers (see figure 3A, switch 82), the at least one switch comprising a

plurality of communication interfaces to establish communication links with the hot plug modules that are coupled to the plurality of couplers (see figure 3A, switch 80 and column 7 lines 28-50), and a plurality of interface control registers to control the plurality of communication interfaces (see column 5 lines 53-60).

As for claim 32, Bealkowski teaches a system management processor to initialize hot plug modules coupled the plurality of couplers (see figure 2, service processor 31 and column 5 lines 30-60).

As for claim 33, Bealkowski teaches wherein the plurality of interface control registers track states of the plurality of communication interfaces and associated hot plug modules (see column 5 lines 30-60).

As for claim 34, Bealkowski teaches wherein the switch provides an indication as to when all pending transactions associated with a hot plug module to be removed have completed (see figure 4, step 130 and column 9 lines 9-32).

Response to Arguments

1. In response to the applicant's arguments that Bealkowski fails to teach limitation in claim 1 for enabling a communication interface of the processor cards 11a-11d. Claim 1 indicated "enabling a communication interface of the hot plug module to establish a communication link with the running computing device". This phrase clearly does not require a communication interface on the processor cards 11a-11d as applicant argues. Figure 1, 3A-3B to Bealkowski clearly discloses the CPU connectors

for the processor cards 11, 20 to plug into the data processing system 10 and therefore, each processor cards 11a-d must have an interface in order to communicate with the data processing system. For example, Bealkowski teaches initialization routines between the processor cards 11, 20 and the data processing system 10 for establishing data communication by determining whether each processor card 11a-d is mated with the CPU connectors correspondingly, which disclosed in figure 4A-4B and column 3 lines 35-41. This response also applies for rejected claims 24 and 31.

2. In response to the applicant's argument for claims 7-8 that Bealkowski fails to teach memory caching to increase the storage capacity. Nowhere in claims 7-8 does the applicant recite adding memory to increase the storage capacity as applicant argues. Bealkowski discloses processor cards are hot plugged to the CPU connectors and each processor card comprising L2 cache as discloses in column 2 lines 42-47. Further, figure 2, Bealkowski teaches a service processor 31 and column 5 lines 31-60, wherein the service processor 31 controls the hot plug controller and also monitoring the events in the computer such as insertion or removal and then stores such information to its own associated memory and controller routine. Furthermore, column 3 lines 35-41, teaches integrated processor cards which including cache into the data processing. Therefore, the caches of the processor cards are considered adding memory to the data processing, which is equivalent to what is claimed "adding the identified memory of the hot plug module to a memory pool of the computing device".

3. In response to the applicant's arguments that Bealkowski fails to teach plurality of interface control registers in response to the hot plug event as claimed 16. Column 5

lines 31-60 to Bealkowski teaches a service processor 31 controls the hot plug controller and also monitoring the events in the computer such as insertion or removal and then stores such information to its own associated memory and controller routine.

4. In response to the applicant's arguments that Bealkowski fails to teach midplane as claimed in claim 28 having a hot plug interface and a hot plug module to update the state of the hot plug interface of the midplane to indicate when the resources are ready to join. Figure 1, Bealkowski teaches the front bus 18 (midplane) having the CPU connectors (hot plug interface) and processor cards 11, 20 (hot plug module) see figure 1 column 3 lines 9-15 and column 6 lines 18-21). Further, Bealkowski teaches the service processor 31 controls the hot plug controller and also monitoring the events in the computer such as insertion or removal and then stores such information to its own associated memory and controller routine as discloses in column 5 lines 31-60.

5. In response to the applicant's arguments that Bealkowski fails to teach a switch that comprises a plurality of interface control registers. Figure 3A to Bealkowski teaches the Power FET switches 82, 86 comprises clock buffer for controlling clock sources when a processor card is removed from the CPU connector (column 6 lines 46-53), Bealkowski also teaches the service processor 31 has its own associated memory for controlling routines such as initialization routines to enable communication for the hot plugged processor card (see column 3 lines 35-41) and recording hardware monitoring, logging events, and reporting on operating conditions within data system 10 (column 5 lines 53-60).

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

TC 2100 director's approval for new ground of rejection

Tim Vo

TIM VO
PRIMARY EXAMINER

Conferees:

Lynne Browne

LYNNE H. BROWNE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Rehana Perveen

REHANA PERVEEN
PRIMARY EXAMINER
SPE, TC2100